

## APPENDIX

### List of Contents

5	Attachment 1	4 pages
	Attachment 2	16 pages

2025-06-10 09:44:00

[illegible]

```

;*****
TITLE "autonomous sine wave tone generator"
LIST P=16C621A, R=DEC

INCLUDE <P16C621A.INC>
__CONFIG __BODEN_OFF&_CP_OFF&_PWRTE_ON&_WDT_OFF&_HS_OSC

;
;*****
; File: TONE.ASM
; Author: Jeremy Sommer
; Date: 07/05/00
; Assembler: MPASM V01.40
; Xtal: 4.14 Mhz
; Inst Clk: 1.035 Mhz (966.2 nSec)
;*****
; Description:
;
; ROM Usage: words
;
; RAM Usage: bytes
;
;***** Constant Definition *****
STEP# EQU .8 ; Number of steps

;***** Register Definition *****
COMP_FLAG EQU 0x20 ; COMPARATOR OUTPUTS
REP_ID EQU 0x21 ; REPEATER ID (0x00 = CPE end)

;***** Bit Definition *****
SIGCOMP_OUT EQU .6 ; Signal comparator output
VCOMP_OUT EQU .6 ; Supply Voltage comparator output
IZCOMP_OUT EQU .7 ; Shunt Current comparator output
TEMPCOMP_OUT EQU .7 ; Temperature comparator output
RB1 EQU .1 ; PORT_B<1>

;*****
; Reset Vector
;*****
org 0x000
goto Start ; Begining of Program

;*****
; Main Routine
;*****
Start
bcf STATUS,RP0 ; Select bank 0
clrf PORTA ; Initialize Port_A by setting output latches
clrf PORTB ; Initialize Port_B by setting output latches

bsf STATUS,RP0 ; Select register bank 1
movlw 0xDF ; Configure TMR0 to run off Fclk/4 (for debugging)
movwf OPTION_REG ;

```

```

    movlw 0x88          ; Configure for internal voltage reference of 2.5V nom.
    movwf VRCON         ;
    movlw 0x1F          ; Set Port_A comparator as inputs
    movwf TRISA         ;
    movlw 0xFF          ; Set Port_B as inputs
    movwf TRISB         ;

    bcf STATUS,RP0      ; Select register bank 0
    movf PORTB,0         ; Identify repeater {0xFF = CO end of 2, 0x00 =
CPE end}
    movwf REP_ID        ;
    movlw 0x02          ; Configure for 4 inputs muxed to 2 comparators
    movwf CMCON         ; Enable pins for I/O functions
    movlw 0xFF          ; Set Port_B to 0xFF (.255)
    movwf PORTB         ;

    bsf STATUS,RP0      ; Select register bank 1
    movlw 0x00          ; Set Port_B as outputs
    movwf TRISB         ;

    bcf STATUS,RP0      ; Select register bank 0 (Initialization complete)

HealthCheck
                                ; (Placeholder for health verification via comparators)

SelectTone
    bcf CMCON,CIS        ; Select Signal and Supply Voltage comparators
    btfss REP_ID,RB1     ; If REP_ID<1> is 0 (CPE end repeater),
    goto Tone5_Loop      ; test for tone5 generation
;    goto Tone4_Loop     ; else test for tone4 generation
                                ; (commented out due to sequentiality)

Tone4_Loop
    btfsc CMCON,SIGCOMP_OUT ; If Signal comparator is high (low signal power),
    goto Tone4_Loop      ; stay static (no tone)

tone4                          ; 17.25 kHz sine at 2/60 UI intervals; 4 instructions
overhead
    movlw .247             ; 4
    movwf PORTB            ;
    movlw .233             ; 6
    movwf PORTB            ;
    movlw .215             ; 8
    movwf PORTB            ;
    movlw .193             ; 10
    movwf PORTB            ;
    movlw .168             ; 12
    movwf PORTB            ;
    movlw .141             ; 14
    movwf PORTB            ;
    movlw .114             ; 16
    movwf PORTB            ;
    movlw .87              ; 18
    movwf PORTB            ;
    movlw .62              ; 20
    movwf PORTB            ;
    movlw .40              ; 22

```

```

movwf PORTB      ;
movlw  .22        ; 24
movwf PORTB      ;
movlw  .8          ; 26
movwf PORTB      ;
movlw  .0          ; 28
movwf PORTB      ;
movlw  .0          ; 30
movwf PORTB      ;
movlw  .0          ; 32
movwf PORTB      ;
movlw  .8          ; 34
movwf PORTB      ;
movlw  .22         ; 36
movwf PORTB      ;
movlw  .40         ; 38
movwf PORTB      ;
movlw  .62         ; 40
movwf PORTB      ;
movlw  .87         ; 42
movwf PORTB      ;
movlw  .114        ; 44
movwf PORTB      ;
movlw  .141        ; 46
movwf PORTB      ;
movlw  .168        ; 48
movwf PORTB      ;
movlw  .193        ; 50
movwf PORTB      ;
movlw  .215        ; 52
movwf PORTB      ;
movlw  .233        ; 54
movwf PORTB      ;
movlw  .247        ; 56
movwf PORTB      ;
movlw  .255        ; 58
movwf PORTB      ;

```

```

goto Tone4_Loop ; Go back to recheck signal level

```

Tone5\_Loop

```

    btfsc CMCON,SIGCOMP_OUT ; If Signal comparator is high (low signal power),
    goto Tone5_Loop ; stay static (no tone)

```

tone5 ; 21.5625 kHz sine at 2/48 UI intervals; 4 instructions overhead

```

movlw  .242        ; 4
movwf PORTB      ;
movlw  .221        ; 6
movwf PORTB      ;
movlw  .193        ; 8
movwf PORTB      ;
movlw  .162        ; 10
movwf PORTB      ;
movlw  .127        ; 12
movwf PORTB      ;

```

```

movlw  .93      ; 14
movwf  PORTB    ;
movlw  .62      ; 16
movwf  PORTB    ;
movlw  .34      ; 18
movwf  PORTB    ;
movlw  .13      ; 20
movwf  PORTB    ;
movlw  .0       ; 22
movwf  PORTB    ;
movlw  .0       ; 24
movwf  PORTB    ;
movlw  .0       ; 26
movwf  PORTB    ;
movlw  .13      ; 28
movwf  PORTB    ;
movlw  .34      ; 30
movwf  PORTB    ;
movlw  .62      ; 32
movwf  PORTB    ;
movlw  .93      ; 34
movwf  PORTB    ;
movlw  .128     ; 36
movwf  PORTB    ;
movlw  .162     ; 38
movwf  PORTB    ;
movlw  .193     ; 40
movwf  PORTB    ;
movlw  .221     ; 42
movwf  PORTB    ;
movlw  .242     ; 44
movwf  PORTB    ;
movlw  .255     ; 46
movwf  PORTB    ;

goto  Tone5_Loop ; Go back to recheck signal level

END          ; That's all Folks !

```

0988333-0644

# Attachment 2

```

;*****
TITLE "non-autonomous sine wave tone generator"
LIST P=16C621A, R=DEC

INCLUDE <P16C621A.INC>
__CONFIG __BODEN_ON&_CP_OFF&_PWRTE_ON&_WDT_OFF&_HS_OSC
;
;*****
; File: TONE.ASM
; Author: Jeremy Sommer
; Date: 07/05/00
; Assembler: MPASM V01.40
; Xtal: 16.56 Mhz
; Inst Clk: 4.14 Mhz (241.5 nSec)
;*****
; Description:
;
; ROM Usage: words
;
; RAM Usage: bytes
;
;***** Constant Definition *****
REP_IDMASK# EQU 0xC0 ; Mask of CMD and PORT_B bits which select
; repeater ID
CO_ID# EQU 0x40 ; REP_ID for CO end of multiple repeaters
CPE_ID# EQU 0xC0 ; REP_ID for CPE end
VRMASK# EQU 0x2F ; Mask of valid VAL bits to use for Vref
CMODE# EQU 0x02 ; Comparator mode for all four comparators
COUTMASK# EQU 0xC0 ; Comparator output bits
HI_TEMP# EQU 0x8D ; High temperature health threshold -> 82C
HI_SUPPLY# EQU 0x8C ; High supply voltage health threshold -> 34.4V
LO_SUPPLY# EQU 0x87 ; Low supply voltage health threshold -> 25.8V
HI_IZ# EQU 0x8D ; High shunt current health threshold ->
32.8mA
LO_IZ# EQU 0xA1 ; Low shunt current health threshold ->
2.1mA
PRESCALE# EQU .0; .7 ; Prescale 2^(7+1)=256
INIT_TMSB# EQU .3; .15 ; Initial TIME_MSB
NSAMPLES# EQU .4; .127 ; Number of samples per burst
NMATCHING# EQU .3; .102 ; Minimum number of matching samples
; (should be set to 80% of NSAMPLES#)
ADDRN_A# EQU 0x3A ; Latest Phase A sample register
ADDRN_B# EQU 0x4A ; Latest Phase B sample register
ADDRN# EQU 0x5A ; Latest temporary sample register
NUM_ADDRS# EQU 0x0A ; Number of sample registers per phase
; (from first bits received)
PAT12# EQU 0x54 ; Pattern1 MSBs minus 1
PAT11# EQU 0xAB ; Pattern1 ... minus 1
PAT10# EQU 0x54 ; Pattern1 LSBs minus 1
PAT22# EQU 0xA9 ; Pattern2 MSBs minus 1
PAT21# EQU 0x52 ; Pattern2 ... minus 1
PAT20# EQU 0xA9 ; Pattern2 LSBs minus 1
; (to last bits received)

;***** Register Definition *****

```

T04T30" 8822330







Table 1. Demographic characteristics of the study population	
Age (years)	50.0 ± 10.0
Gender	
Male	50.0%
Female	50.0%
Education (years)	12.0 ± 2.0
Marital status	
Married	80.0%
Single	20.0%
Occupation	
Professional	30.0%
Managerial	20.0%
Technical	10.0%
Service	20.0%
Unemployed	20.0%
Income (USD/month)	1,200 ± 300
Health status	
Good	70.0%
Fair	20.0%
Poor	10.0%

```

    bcf    STATUS,RP0    ; Select register bank 0
    movlw  CO_ID#        ; Prepare to set REP_ID to CO_ID#
    btfss  PORTB,7       ; If PORT_B<7> = 0 (indicates the CPE end
repeater)
    movlw  CPE_ID#       ; Prepare to set REP_ID to CPE_ID#
    movwf  REP_ID        ; Set REP_ID
    movlw  0x02          ; Configure for 4 inputs muxed to 2 comparators
    movwf  CMCON         ; Enable pins for I/O functions
    movlw  0xFF          ; Set Port_B to 0xFF (.255)
    movwf  PORTB         ;

    bsf    STATUS,RP0    ; Select register bank 1
    movlw  0x01          ; Set Port_B as outputs except for RB0, which is the INT
input
    movwf  TRISB         ;

    bcf    STATUS,RP0    ; Select register bank 0 (Initialization complete)
    goto   TestHealth    ; Verify card health, generate ACK tone if healthy

```

```

TakeSample                ; TAKE SAMPLE BURST,
                          ; VALIDATE CORRELATION,
                          ; SHIFT INTO SAMPLE REGISTER FOR CURRENT PHASE,
                          ; TRANSFER TO TEMPORARY SAMPLE REGISTER
    comf    PHASE,1        ; switch phase of sampling (alternate A and B)
    call    GetSamples     ; accumulate NSAMPLES# samples
    movwf   TEMP1         ; recover sample value (0=0,1=1,.255=invalid)
    btfsc   TEMP1,7        ; If invalid sample,
    goto    ResetCurrentPhase ; clear sample registers of current phase
                          ; and start waiting all over again
    movlw   ADDR_N_A#      ; Starting with latest Phase A register
    btfsc   PHASE,0        ; If PHASE = 1 (B)
    movlw   ADDR_N_B#      ; switch to latest Phase B register

```



```

movlw PAT22#           ; Put PAT22# into W register
subwf PAT22,1          ; Subtract PAT22# from PAT22
decfsz PAT22,1         ; If pattern did not match
goto InfiniteLoop      ; wait indefinitely
movlw PAT21#           ; Put PAT21# into W register
subwf PAT21,1          ; Subtract PAT21# from PAT21
decfsz PAT21,1         ; If pattern did not match
goto InfiniteLoop      ; wait indefinitely
movlw PAT20#           ; Put PAT20# into W register
subwf PAT20,1          ; Subtract PAT20# from PAT20
decfsz PAT20,1         ; If pattern did not match
goto InfiniteLoop      ; wait indefinitely
                        ; Pattern matches!

```

CheckOverallParity ; CHECK CMD AND VAL BITS FOR PARITY

```

movf CMD1,0            ; Put CMD1 into W register
call CheckParity       ; Check that bits 7,6 and 3,2
                        ; are complements of bits 5,4
                        ; and 1,0 respectively
movwf TEMP1            ; Recover return value
btfsc TEMP1,7          ; If return value indicates invalid parity
goto InfiniteLoop      ; wait indefinitely
movf CMD0,0            ; Put CMD0 into W register
call CheckParity       ; Check that bits 7,6 and 3,2
                        ; are complements of bits 5,4
                        ; and 1,0 respectively
movwf TEMP1            ; Recover return value
btfsc TEMP1,7          ; If return value indicates invalid parity
goto InfiniteLoop      ; wait indefinitely
movf VAL1,0            ; Put VAL1 into W register
call CheckParity       ; Check that bits 7,6 and 3,2
                        ; are complements of bits 5,4
                        ; and 1,0 respectively
movwf TEMP1            ; Recover return value
btfsc TEMP1,7          ; If return value indicates invalid parity
goto InfiniteLoop      ; wait indefinitely
movf VAL0,0            ; Put VAL0 into W register
call CheckParity       ; Check that bits 7,6 and 3,2
                        ; are complements of bits 5,4
                        ; and 1,0 respectively
movwf TEMP1            ; Recover return value
btfsc TEMP1,7          ; If return value indicates invalid parity
goto InfiniteLoop      ; wait indefinitely
                        ; Parity is OK!

```

InterpretCmdVal ; INTERPRET CMD AND VAL BITS FOR ACTION

```

clrf TEMP2             ; Clear TEMP2
movf VAL1,0            ; Put VAL1
movwf TEMP1            ; into TEMP1
call ConcatenateBits   ; Shift the concatenated command bits
                        ; into TEMP2
movf VAL0,0            ; Put VAL0
movwf TEMP1            ; into TEMP1
call ConcatenateBits   ; Shift the concatenated command bits
                        ; into TEMP2
movf TEMP2,0           ; Put TEMP2
movwf VAL              ; into VAL

```









```

        movwf CMCON          ;
;*****
****
        return              ; return from DefaultComparators

PauseN
        movwf TEMP2          ; Put W register in TEMP2
PauseCycle
        decfsz    TEMP2,1      ; If not done counting down,
        goto     PauseCycle    ; keep counting down
        return                ; else return

ConcatenateBits
; Shift TEMP1<7,6> into TEMP2<1,0>
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
        rlf      TEMP2,1      ; Rotate TEMP2 left through carry
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
        rlf      TEMP2,1      ; Rotate TEMP2 left through carry
; Shift TEMP1 left twice
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
; Shift TEMP1<3,2> into TEMP2<1,0>
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
        rlf      TEMP2,1      ; Rotate TEMP2 left through carry
        rlf      TEMP1,1      ; Rotate TEMP1 left through carry
        rlf      TEMP2,1      ; Rotate TEMP2 left through carry
        return                ; return from ConcatenateBits

CheckParity
        movwf TEMP1          ; Move W register into TEMP1
        rrf      TEMP1,1      ; Rotate right two bits
        rrf      TEMP1,1      ; through carry
        xorwf    TEMP1,1      ; XOR with original TEMP1
        btfss    TEMP1,0      ; If bit 0 parity is incorrect
        retlw    .255         ; Return value .255
        btfss    TEMP1,1      ; If bit 1 parity is incorrect
        retlw    .255         ; Return value .255
        btfss    TEMP1,4      ; If bit 4 parity is incorrect
        retlw    .255         ; Return value .255
        btfss    TEMP1,5      ; If bit 5 parity is incorrect
        retlw    .255         ; Return value .255
        retlw    .0           ; Return value .0

GetSamples
        clrf     COUNT1        ; set COUNT1=0
        movlw    NSAMPLES#     ; initialize sample counter COUNT to NSAMPLES#
        movwf    COUNT         ;

StartSampling
;*****
***
        ;btfsc    CMCON,SIGCOMP_OUT ; if Signal Comparator = 1 (high amplitude)
        btfsc    PORTA,RA0      ; DEBUG - use RA0 instead
;*****
***
        incf     COUNT1,1      ; Increment COUNT1
        decfsz    COUNT,1      ; Decrement COUNT
; PLACEHOLDER for additional delay between samples

```







```

movwf PORTB      ;
movlw  .31        ; 37
movwf PORTB      ;
movlw  .21        ; 39
movwf PORTB      ;
movlw  .13        ; 41
movwf PORTB      ;
movlw  .7         ; 43
movwf PORTB      ;
movlw  .2         ; 45
movwf PORTB      ;
movlw  .0         ; 47
movwf PORTB      ;
movlw  .0         ; 49
movwf PORTB      ;
movlw  .2         ; 51
movwf PORTB      ;
movlw  .7         ; 53
movwf PORTB      ;
movlw  .13        ; 55
movwf PORTB      ;
movlw  .21        ; 57
movwf PORTB      ;
movlw  .31        ; 59
movwf PORTB      ;
movlw  .43        ; 61
movwf PORTB      ;
movlw  .57        ; 63
movwf PORTB      ;
movlw  .71        ; 65
movwf PORTB      ;
movlw  .86        ; 67
movwf PORTB      ;
movlw  .103       ; 69
movwf PORTB      ;
movlw  .119       ; 71
movwf PORTB      ;
movlw  .136       ; 73
movwf PORTB      ;
movlw  .152       ; 75
movwf PORTB      ;
movlw  .169       ; 77
movwf PORTB      ;
movlw  .184       ; 79
movwf PORTB      ;
movlw  .198       ; 81
movwf PORTB      ;
movlw  .212       ; 83
movwf PORTB      ;
movlw  .224       ; 85
movwf PORTB      ;
movlw  .234       ; 87
movwf PORTB      ;
movlw  .242       ; 89
movwf PORTB      ;
movlw  .248       ; 91
movwf PORTB      ;

```

<p>                     1. <i>Staphylococcus aureus</i> (ATCC 12228)                      2. <i>Staphylococcus aureus</i> (ATCC 12228)                      3. <i>Staphylococcus aureus</i> (ATCC 12228)                      4. <i>Staphylococcus aureus</i> (ATCC 12228)                      5. <i>Staphylococcus aureus</i> (ATCC 12228)                      6. <i>Staphylococcus aureus</i> (ATCC 12228)                      7. <i>Staphylococcus aureus</i> (ATCC 12228)                      8. <i>Staphylococcus aureus</i> (ATCC 12228)                      9. <i>Staphylococcus aureus</i> (ATCC 12228)                      10. <i>Staphylococcus aureus</i> (ATCC 12228)                 </p>	
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100
9	100
10	100
11	100
12	100
13	100
14	100
15	100
16	100
17	100
18	100
19	100
20	100
21	100
22	100
23	100
24	100
25	100
26	100
27	100
28	100
29	100
30	100
31	100
32	100
33	100
34	100
35	100
36	100
37	100
38	100
39	100
40	100
41	100
42	100
43	100
44	100
45	100
46	100
47	100
48	100
49	100
50	100
51	100
52	100
53	100
54	100
55	100
56	100
57	100
58	100
59	100
60	100
61	100
62	100
63	100
64	100
65	100
66	100
67	100
68	100
69	100
70	100
71	100
72	100
73	100
74	100
75	100
76	100
77	100
78	100
79	100
80	100
81	100
82	100
83	100
84	100
85	100
86	100
87	100
88	100
89	100
90	100
91	100
92	100
93	100
94	100
95	100
96	100
97	100
98	100
99	100
100	100

```
tone15          ; 64.6875 kHz sine at 2/64 UI intervals; 2
instructions overhead
```

```

movlw    .250                ; 3
movwf    PORTB              ;
movlw    .240                ; 5
movwf    PORTB              ;
movlw    .227                ; 7
movwf    PORTB              ;
movlw    .209                ; 9
movwf    PORTB              ;
movlw    .188                ; 11
movwf    PORTB              ;
movlw    .165                ; 13
movwf    PORTB              ;
movlw    .140                ; 15
movwf    PORTB              ;
movlw    .115                ; 17
movwf    PORTB              ;
movlw    .90                 ; 19
movwf    PORTB              ;
movlw    .67                 ; 21
movwf    PORTB              ;
movlw    .46                 ; 23
movwf    PORTB              ;
movlw    .28                 ; 25
movwf    PORTB              ;
movlw    .15                 ; 27
movwf    PORTB              ;
movlw    .5                  ; 29
movwf    PORTB              ;
movlw    .0                  ; 31
movwf    PORTB              ;
movlw    .0                  ; 33
movwf    PORTB              ;
movlw    .5                  ; 35
movwf    PORTB              ;
movlw    .15                 ; 37
movwf    PORTB              ;
movlw    .28                 ; 39
movwf    PORTB              ;
movlw    .46                 ; 41
movwf    PORTB              ;
movlw    .67                 ; 43
movwf    PORTB              ;
movlw    .90                 ; 45
movwf    PORTB              ;
movlw    .115                ; 47
movwf    PORTB              ;
movlw    .140                ; 49
movwf    PORTB              ;

```

